What is claimed is:

1. An encoding method for compressing data, the method comprising the steps of:

   (a) encoding the data to produce encoded data;

   (b) forming difference data between the data and the encoded data;

   (c) dividing the difference data into one or more primary blocks, forming difference blocks;

   (d) using a selected codebook re-encoding a difference block to produce an encoded difference block;

   (e) calculating a following difference block between said difference block and the encoded difference block, forming secondary difference blocks;

   (f) iteratively repeating steps d, e and f, until a desired level of compression is achieved;

   (g) wherein said step f is preformed for a plurality of selected primary and secondary difference blocks;

   (h) wherein the codebook for said step d is selected for each iteration from a plurality of codebooks, at least one of said codebooks contains codevectors trained with training difference material, wherein prior to the training, said training difference material is preprocessed for individually adapting frequency distribution of at least one of said codevectors for weighting to particular portions of the data.

2. A method according to claim **1**, wherein at least at one of said repetitions the difference blocks are divided into sub-blocks at least one of which to be used as difference blocks at a subsequent repetition.

3. A method according to claim **1**, wherein a plurality of codebooks are used in said step d.

4. A method according to claim **2**, wherein a plurality of codebooks are used in said step d.

5. A method according to claim **1**, wherein the preprocessing of the training material is made using discrete cosine transform.

6. A method according to claim **1**, wherein the preprocessing of the training material is made using a functional transform.

7. A method according to claim **1**, further comprising the steps of evaluating the cost of a repetition using a cost function which produces a cost result, and deciding if to perform the next repetition based on the basis of said result.

8. A method according to claim **7**, wherein the cost function utilizes a remaining difference, and a number of bits used for representing said difference block, to calculate a cost of further repetitions.

9. A method according to claim **8**,wherein the number of bits is weighted.

10. A method according to claim **2**, wherein the preprocessing of the training material is made using discrete cosine transform.

11. A method according to claim **2**, wherein the preprocessing of the training material is made using a functional transform.

12. A method according to claim **2**, further comprising the steps of evaluating the cost of a repetition using a cost function which produces a cost result, and deciding if to perform the next repetition based on the basis of said result.

13. A method according to claim **12**, wherein the cost function utilizes a remaining difference, and a number of bits used for representing said difference block, to calculate a cost of further repetitions.

==24==

14. A method according to claim **13**, wherein the number of bits is weighted.

15. A method according to claim **3**, wherein the preprocessing of the training material is made using discrete cosine transform.

16. A method according to claim **3**, wherein the preprocessing of the training material is made using any functional transform.

17. A method according to claim **3**, further comprising the steps of evaluating the cost of a repetition using a cost function which produces a cost result, and deciding if to perform the next repetition based on the basis of said result.

18. A method according to claim **17**, wherein the cost function utilizes a remaining difference, and a number of bits used for representing said difference block , to calculate a cost of further repetitions.

19. A method according to claim **18**, wherein the number of bits is weighted.

20. A method according to claim **4**, wherein the preprocessing of the training material is made using discrete cosine transform.

21. A method according to claim **4**, wherein the preprocessing of the training material is made using a functional transform.

22. A method according to claim **4**, further comprising the steps of evaluating the cost of a repetition using a cost function which produces a cost result, and deciding if to perform the next repetition based on the basis of said result.

23. A method according to claim **22**, wherein the cost function utilizes a remaining difference, and a number of bits used for representing said difference block, to calculate a cost of further repetitions.

24. A method according to claim **23**, wherein the number of bits is weighted.

25. A method according to claim **1**, wherein at least at one repetition the difference blocks are preprocessed before encoding.

26. A method according to claim **2**, wherein at least at one repetition the difference blocks are preprocessed before encoding.

==25==

27. A decoding method for pre-compressed data, the method comprising the steps of:

producing a plurality of codebooks for the decompression of encoded difference data, wherein at least one of said codebooks contains codevectors, which have been weighted to a specific frequency distribution; and,

decompressing data using the codebooks in combination, to produce a decompression result which comprises at least a plurality of significant frequencies contained in said data prior to compression.

28. An encoder for compressing data, comprising:

a) means for encoding the data;

b) means for forming difference data between the data and the encoded data;

c) means for dividing the difference data into one or more primary blocks, forming the latest difference data blocks;

d) means for iteratively repeating the following step of re-encoding and calculating independently for each block, until a desired accuracy level of compressed data is achieved:

e) means for re-encoding a step-specific difference data block, which is the latest difference data block, using a codebook, elected suitable for each repetition, the codebook for said step-specific difference block containing codevectors;

f) means for calculating a following difference block between the step-specific difference block and the encoded step-specific difference block, forming the latest difference data block;

g) and at least one of said codebooks contains codevectors trained with training difference material, wherein prior the training, said training difference material is preprocessed for individually adapting frequency distribution of each codevector for weighting to particular information of the data.

29. An encoder according to claim **28**, wherein at least at one repetitions the difference blocks are divided into sub-blocks for being used as difference blocks at a subsequent repetitions.

30. An encoder according to claim **28**, wherein said means for re-encoding are adapted to use a plurality of codebooks.

31. An encoder according to claim **29**, wherein said means for re-encoding are adapted to use a plurality of codebooks.

32. An encoder according to claim **28**, wherein the evaluation means further comprise a cost function, which calculates the cost of using the additional repetition.

33. An encoder according to claim **32**, wherein the cost function takes into account a remaining difference, and the number of bits used for representing said difference block, to calculate a cost of further repetitions.

34. An encoder according to claim **33**, wherein the number of bits is weighted.

35. An encoder according to claim **29**, wherein the evaluation means further comprises a cost function, which calculates the cost of using the additional repetition.

36. An encoder according to claim **35**, wherein the cost function the cost function utilizes a remaining difference, and a number of bits used for representing said difference block, to calculate a cost of further repetitions.

37. An encoder according to claim **36**, wherein the number of bits is weighted.

38. An encoder according to claim **30**, wherein the evaluation means further comprises a cost function, which calculates the cost of using the additional repetition.

39. An encoder according to claim **38**, wherein the cost function the cost function utilizes a remaining difference, and a number of bits used for representing said difference block, to calculate a cost of further repetitions.

40. An encoder according to claim **39**, wherein the number of bits is weighted.

41. An encoder according to claim **31**, wherein the evaluation means further comprises a cost function, which calculates the cost of using the additional repetition.

42. An encoder according to claim **41**, wherein the cost function the cost function utilizes a remaining difference, and a number of bits used for representing said difference block, to calculate a cost of further repetitions.

43. An encoder according to claim **42**, wherein the number of bits is weighted.

44 A compressed data decoder for decompression of encoded data, the encoded data containing a plurality of encoded difference data said decoder comprising:

a compressed data input module;

a decompression module adapted to utilize at least one codebook that has been weighted to a specific frequency distribution; and,

a decompressed data output module.

45. A computer readable media containing software that when executed by a computer, will case said computer to perform substantially in accordance with the method of claim **1**.

46. A computer readable media containing software that when executed by a computer, will case said computer to perform substantially in accordance with the method of claim **27**.

47. A computer readable media containing software that when executed by a computer, will case said computer to perform substantially as the encoder of claim **28**.

48. A computer readable media containing software that when executed by a computer, will case said computer to perform substantially as the decoder of claim **44**.